

# 5 虚拟存储器、请求分页

2018年11月12日 8:30



## ◆ 虚拟存储器概述

### 一. 常规存储器管理方式的特征和局部性原理

#### 1. 常规存储器管理方式的特征

- 1) 一次性：把作业全装入内存才能运行，限制了处理机利用率和系统吞吐量
- 2) 驻留性：即使进程阻塞，不会换出任一部分

#### 2. 局部性原理

- 1) 较短时间内，程序执行仅局限于某部分
  - (1) 除少部分转移和调用外，基本是顺序执行
  - (2) 调用函数会使程序运行轨迹转至另一区域，但调用深度一般不超过5
  - (3) 循环结构只有少数指令被多次执行
  - (4) 多数据结构的处理一般局限于很小的范围内
- 2) 时间局限性：短时间内，指令/数据可能被再次执行/访问，如循环
- 3) 空间局限性：短时间内，访问的存储单元一般集中在一定范围内，如顺序

#### 3. 虚拟存储器的基本工作情况

- 1) 若要访问的页/段尚未调入内存，便发出缺页/段中断请求，尝试调入
- 2) 若内存已满，则置换出一部分页/段，再调入进来

### 二. 虚拟存储器的定义和特征

#### 1. 虚拟存储器的定义

- 1) 有**请求调入功能和置换功能**，能逻辑上对内存加以扩充的存储器系统
- 2) 能使用户感觉到的内存容量比实际内存容量大得多
- 3) 逻辑容量取决于内、外存容量之和，运行速度近内存速度，成本近外存
- 4) 内存利用率高，程序并发度高，系统吞吐量大，广泛应用于大中小微型机

#### 2. 虚拟存储器的特征

- 1) 多次性：允许将作业分多次装入内存
- 2) 对换性：允许将暂不使用的代码/数据甚至整个进程调至外存
- 3) 虚拟性：逻辑上扩充内存容量，能运行更大作业，提高多道程序度

#### 3. 多次性和对换性是虚拟性的基础；离散分布是多次性和对换性的基础

### 三. 虚拟存储器的实现方法

#### 1. 分页/段请求系统：在分页/段系统的基础上，**增加了请求调页/段功能和页/段置换功能所形成的虚拟存储系统**（页和段视作两种）

- 1) 允许只装入少数页/段的程序/数据即可启动
- 2) 硬件支持：请求分页/段的页/段表机构、缺页/段中段机构、地址变换机构
  - (1) 不少硬件被集成在处理器芯片上
- 3) 软件支持：实现请求调页/段的软件和实现页/段置换的软件
  - (1) 段的长度可变，因而内存分配和回收更复杂



## ◆ 请求分页存储管理方式

## 一. 请求分页中的硬件支持

### 1. 请求页表机制：

- 1) 

页号	物理块号	状态位 P	访问字段 A	修改位 M	外存地址
----	------	-------	--------	-------	------
- 2) 状态位/存在位 P：指示该页是否已调入内存，供访问时参考
- 3) 访问字段 A：记录本页在一段时间内被访问的次数，或记录本页最近已有多长时间未被访问，供换出时参考
- 4) 修改位 M：表示该页在调入内存后是否被修改过，供置换时参考
- 5) 外存地址：指出该页在外存上的地址，通常是物理块号，供调入时参考

### 2. 缺页中断机构

- 1) 缺页中断作为中断，也需要保护cpu环境、分析中断源、转入中断处理程序、再恢复cpu环境，但也有如下不同于其他中断的特点：

- (1) 在指令执行期间产生和处理中断信号。通常，CPU 都是在一条指令执行完后，才检查是否有中断请求到达。然而，**缺页中断是在指令执行期间**，发现所要访问的指令或数据不在内存时所产生和处理的
- (2) 一条指令在执行期间，可能产生多次缺页中断。如在执行一条指令 COPY A TO B 时，可能要产生 6 次缺页中断，其中指令本身跨了两个页面，A 和 B 又分别各是一个数据块，也都跨了两个页面。基于这些特征，系统中的硬件机构应能保存多次中断时的状态，并保证最后能返回到中断前产生缺页中断的指令处继续执行

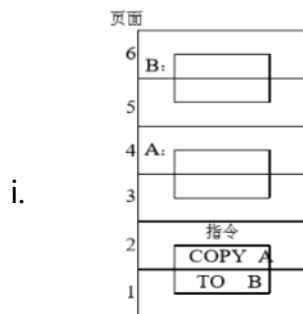


图 4-24 涉及 6 次缺页中断的指令

### 3. 地址变换机构

- 1) 分页系统地址变换机构的基础上，增加了产生和处理缺页中断，以及从内存中换出一页的功能等
  - (1) 先检索快表，找到待访问页后将修改位M置1
  - (2) 若快表中无该页表项，应去内存中找页表
  - (3) 在页表中根据状态位P确认该页在内存中的位置
  - (4) 若确认后发现尚未调入内存，应发生缺页中断，由OS调入

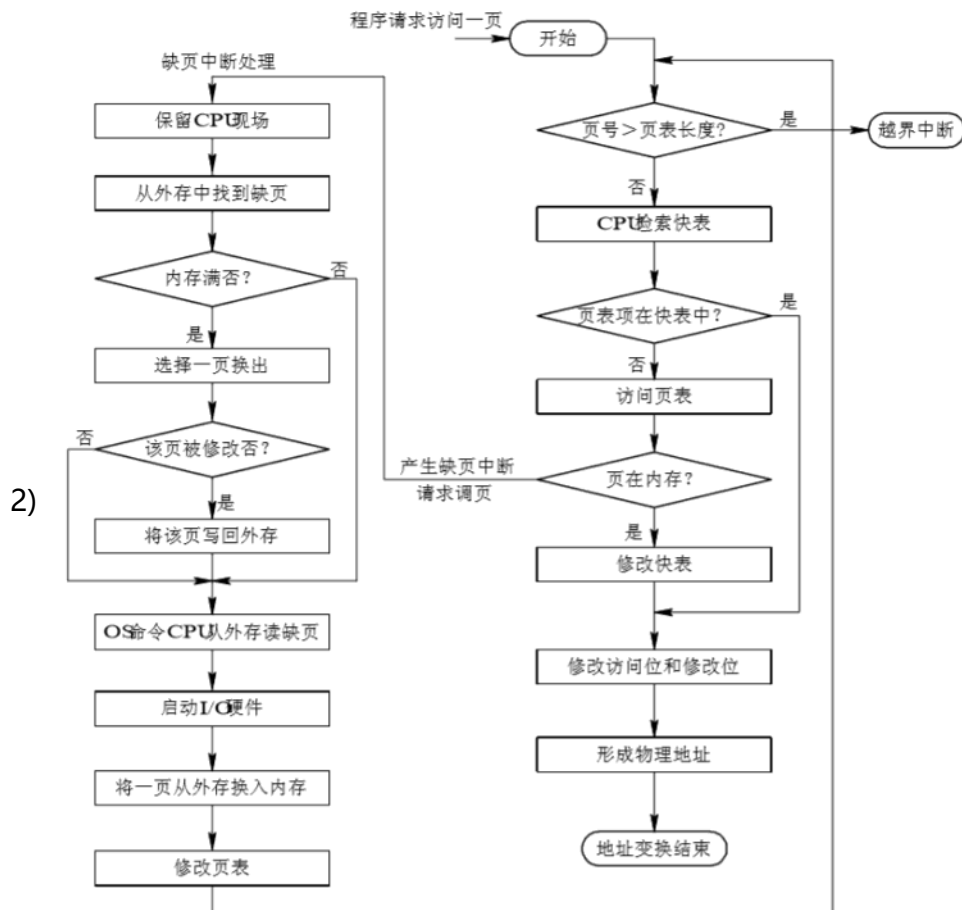


图 4-25 请求分页中的地址变换过程

(1) 左边的都是中断处理，右边的接近传统的变换机构

## 二. 请求分页中的内存分配

### 1. 最小物理块数的确定

- 1) 最小物理块数：能保证进程正常运行所需的最小物理块数
- 2) 按最小物理块数分配资源，很可能使缺页率极高，使执行速度极慢
- 3) 真正分配的物理块数取决于计算机硬件结构、指令格式、功能、寻址方式
  - (1) 单地址指令只需2块，一个存指令。一个存数据
  - (2) 允许间接寻址的机器需要至少3块
  - (3) 指令长度可能有两个字节的，需要6块，见图4-25

### 2. 内存分配策略

- (1) 固定分配：为每个进程分配固定数目物理块，运行期间不再改变
  - (2) 可变分配：先分配一定数目，运行期间适当增减
  - (3) 局部置换：发现缺页后，对页进行对换，确保分配的空间不变
  - (4) 全局置换：发现缺页后，取出一空闲物理块分配给该进程。若无空闲物理块，可能从其他进程选出一页做对换
- 1) 固定分配局部置换(Fixed Allocation, Local Replacement)
    - (1) 一般根据进程类型（交互/批处理）或程序员、管理员的建议来确定
      - i. 若分配太少，会频繁缺页中断，降低系统吞吐量
      - ii. 若分配太多，容易使cpu等资源空闲，且进程对换时更耗时
  - 2) 可变分配全局置换(Variable Allocation, Global Replacement)
    - (1) 一般也是先根据进程类型、程序员、管理员建议来确定

- (2) 全局置换时可能换出其他进程的物理块，使其缺页率增加
- 3) 可变分配局部置换(Variable Allocation, Local Replacement)
  - (1) 先根据进程类型、程序员、管理员建议，分配一些物理块
  - (2) 运行时再根据缺页率高/低，增加/减少一些物理块
- 3. 固定分配策略下的物理块分配算法
  - 1) 平均分配：所有空闲物理块平均分配给各进程
  - 2) 按比例分配：块数=页数/总页数\*总块数（向上取整）
 

系统中各进程页面数的总和为：

$$(1) \quad S = \sum_{i=1}^n S_i$$

假定系统中可用的物理块总数为  $m$ ，则每个进程所能分到的物理块数为  $b_i$

$$(2) \quad b_i = \frac{S_i}{S} \times m$$
  - 3) 考虑优先权：照顾重要的、紧迫的作业分配更多内存
    - (1) 一般是一部分内存按比例分配给各进程，另一部分按优先权分配
    - (2) 实时控制系统可能只按优先权分配物理块

### 三. 页面调入策略

- 1. 何时调入页面
  - 1) 预调页策略：以预测为基础，将不久后可能被访问的页面调入内存
    - (1) 成功率一般只有一半
    - (2) 第一次调入进程时，可由程序员指出需调入的页
    - ☒ (3) 采用工作集的系统，会将工作集中的所有页都调入内存
  - 2) 请求调页策略：发现页不在内存时提出请求，由OS调入
    - (1) 易于实现，广泛采用，但花费较大开销，增加了磁盘I/O启动频率
- 2. 从何处调入页面
  - 1) 将外存分为：存放文件的文件区、存放对换页面的对换区
    - (1) 对换区一般连续分配；文件区一般离散分配
    - (2) 对换区的I/O速度一般比文件区快
  - 2) 从何处调入页面，分以下三种情况
    - (1) 有足够对换区空间：全部调入对换区，提高调页速度
      - i. 运行前需要把相关文件都拷到对换区
    - (2) 缺足够的对换区空间：不会修改的文件从文件区调入
      - i. 因为不修改的文件，在换出时可直接省去，不需修改
      - ii. 但可能修改的部分为了速度，仍需调到对换区
    - (3) UNIX方式：进程相关文件放在文件区；换出页面放在对换区
      - i. 未运行过的页面一般是从文件区调入
      - ii. 运行过（但被换出）的页面从对换区调入
      - iii. 因为UNIX允许页面共享，可能页面已被其他进程调入
- 3. 页面调入过程
  - 1) 存在位P为0，即不在内存时，向CPU发出缺页中断
  - 2) 中断处理程序保留cpu环境，转入缺页中断处理程序
  - 3) 找到所在物理块后，启动I/O调入内存，修改页表
  - 4) 若内存已满，按置换算法选出一页准备换出

- 5) 若换出页修改位M不是0, 即被修改过, 则需要将其数据写回磁盘
- 6) 调入所缺页后将页表中其存在位P改为0, 并加入快表,
- 7) 之后才能根据物理地址访问内存数据, 这个过程对用户透明

#### 4. 缺页率f

- 1) 设逻辑空间页数n, 分配到的内存物理块数m, 访问页面成功 (在内存) 次数S, 访问页面失败次数F, 总访问次数 $A=S+F$ , 则 $f=F/A$
- 2) 影响因素
  - (1) 页面越小, 缺页率越高
  - (2) 分配到的物理块越少, 缺页率越高
  - (3) 页面置换算法越差, 缺页率越高。缺页率是衡量该算法的重要指标
  - (4) 程序固有特性: 局部化程度越低, 缺页率越高
- 3) 根据换出页被修改与否的概率, 可按0-1分布数学期望计算处理时间
  - i.
  - ii.
  - iii.
  - iv.
  - v.
  - vi.
  - vii.
  - viii.
  - ix. -----我是底线-----